DEC STD 074-8 Printer Systems Standards - Tabs

DOCUMENT IDENTIFIER: A-DS-EL00074-08-0000 Rev A1, 20-Sep-1991

ABSTRACT: This section of DEC STD 074 defines tabulation (tab) control functions for Digital printers, including horizontal and vertical tab controls and settings.

APPLICABILITY: The requirements of this document are mandatory for engineers designing hardware for terminal products and Software Engineers designing programs using terminal interfaces.

STATUS: APPROVED 20-Sep-1991; use VTX SMC for the current status.

This document is confidential and proprietary, and is the property of Digital Equipment Corporation. It is an unpublished work protected under the Federal copyright laws.

© Digital Equipment Corporation. 1990, 1991. All rights reserved.

DEC STD 074-8 Printer Systems Standards - Tabs

DOCUMENT IDENTIFIER: A-DS-EL00074-08-0000 Rev A1, 20-Sep-1991

REVISION HISTORY:	Rev A,	15-May-1990	
	Rev A1,	20-Sep-1991	ECO Number NRO01

Document Management Category:	Terminal Interface Architecture (STI)
Responsible Department:	VIPS Hardcopy Engineering
Responsible Person:	Tim Lasko
SMC Writer:	Eric Falkof

APPROVAL: This document has been reviewed and recommended for approval by the general review group for its category and by the Technical Director of Terminals and Systems Architecture. Rev A1 incorporates clerical changes only.

Timothy Lasko - Standard Owner

Eric Williams - Standards Process Manager

Direct requests for further information to:

Timothy Lasko
Use $ VTX ELF for the latest location information.

Use VTX SMC to order copies of this document from Standards and Methods Control. Send distribution questions to JOKUR::SMC or call DTN: 234-4423.

The DIGITAL logo is a trademark of Digital Equipment Corporation.

CONTENTS

1 INTRODUCTION 1

2 GOALS 1

3 INITIAL STATES 2

4 GENERAL CONSIDERATIONS 4
4.1 ALGORITHM FOR HORIZONTAL AND VERTICAL TABS 4
4.2 DEPENDENCIES 7
4.2.1 Tab Stop Distances 7
4.2.2 Tab Stop Scaling 7
4.2.3 HORIZONTAL TAB (HT) Past Last Horizontal Tab Stop 7
4.2.4 VERTICAL TAB (VT) Past Last Vertical Tab Stop 7
4.2.5 Position of Tab Stops Relative to the Character Box 7
4.2.6 Internal Representation Of Tabs Stops 8

5 HORIZONTAL AND VERTICAL TABULATION 9
5.1 HT - HORIZONTAL TABULATION 9
5.2 VT - VERTICAL TABULATION 10

6 SETTING HORIZONTAL AND VERTICAL TABULATION STOPS 12
6.1 HTS - HORIZONTAL TABULATION SET 12
6.2 VTS - VERTICAL TABULATION SET 12
6.3 DECHTS - HORIZONTAL TABULATION SET 13
6.4 DECVTS - VERTICAL TABULATION SET 14
6.5 DECSHTS - SET HORIZONTAL TABULATION STOPS 15
6.6 DECSVTS - SET VERTICAL TABULATION STOPS 16

7 CLEARING HORIZONTAL AND VERTICAL TABULATION STOPS 17
7.1 TBC - TABULATION CLEAR 17
7.2 DECCAHT - CLEAR ALL HORIZONTAL TABULATION STOPS 19
7.3 DECCAVT - CLEAR ALL VERTICAL TABULATION STOPS 19

Appendix A REFERENCED DOCUMENTS 21

A.1 EL-Class Digital Documents 21

A.2 Other Documents 21

A.3 Ordering Information 21

1 INTRODUCTION

Tabulation (tab) settings provide page reference points to enable the alignment of data–in rows and columns, using both horizontal tabs and vertical tabs for rapid positioning.

This section of DEC STD 074 is supported by *DEC STD 074-6 Printer Systems Standard - Page Coordinate System (In process)*

The tab functions defined in this document are organized into three groups.

1. Moving:
   a. HORIZONTAL TABULATION (HT) and VERTICAL TABULATION (VT) are ANSI/ISO C0 control characters that move the active position to the next tab stop.
2. Setting:
   a. HORIZONTAL TABULATION SET (HTS) and VERTICAL TABULATION SET (VTS) set tab stops at the current position. DECHTS and DECVTS are Digital private control functions that are required for compatibility with older Digital products, but are not to be used by conforming software.
   b. SET HORIZONTAL TABULATION STOPS (DECSHTS) and SET VERTICAL TABULATION STOPS (DECSVTS) are Digital private control sequences that add up to 16 additional tab stops to the current list.
3. Clearing:
   a. TABULATION CLEAR (TBC) is the ANSI/ISO control sequence to clear all horizontal and/or vertical tab stops.
   b. CLEAR ALL HORIZONTAL TABULATION STOPS (DECCAHT) and CLEAR ALL VERTICAL HORIZONTAL TABULATION STOPS (DECCAVT) are Digital private escape sequences that are required for compatibility with Digital's older products. Conforming software should use TBC instead.

2 GOALS

The goals of the tab control functions are to:

a. Provide a means to line up data horizontally and/or vertically.

b. Provide a means to represent tabular material in a more compressed form than a series of SPACEs and/or LINE FEEDs.

c. Provide a means of adjustment to fit any application, but have a standard initial setting. At Digital, the standard setting is every 8 character positions.

d. Always move some distance so that consecutive tab characters advance the active position past the next available tab stop, except when the active position is at the right margin.

e. Support both monospaced and proportional spaced characters.

3 INITIAL STATES

Upon power up and after SOFT TERMINAL RESET (DECSTR) and RESET TO INITIAL STATE (RIS), the tab stops shall be initialized as follows.

a. Horizontal tab stops shall be set every 8 character positions from the first position, at the left edge of character positions 9, 17, 25, 34, 41, 49, 57, 65, 73, 81, 89, 97, 105, 113, 121, and 129.

   If the initial page orientation is landscape, because the initial Line Home Position and the initial Left Margin differ, the initial horizontal tab stops shall be set at the left edge of character positions 6, 14, 22, and the like. Refer to DEC STD 074-6 for more information.

   Extra tabs must be set in the tabs list to allow tabs to be set for 132 characters when scaling with SET HORIZONTAL PITCH (DECSHORP).

b. Vertical tabs shall be set for every line position from 1 through 66.

The initial tab stops are selected to be compatible with a portrait orientation, 80 character, 66 line page.

Algorithm:

```
boolean_type first_character_flag;

int ssu;
int pum;
int minimum_horizontal_tab_stops = 32;
int minimum_vertical_tab_stops = 66;
int available_horizontal_stops;
int available_vertical_stops;
int maximum_horizontal_tab_stops;
int maximum_vertical_tab_stops;
int new_horizontal_tab_stop_list[32];
int new_vertical_tab_stop_list[66];

int initialize_horizontal_tab_stops()
{
int i;

i = 1;

available_horizontal_stops = minimum_horizontal_tab_stops;
maximum_horizontal_tab_stops = minimum_horizontal_tab_stops;

horizontal_tab_stop_list[i] = 9;
while(i < maximum_horizontal_tab_stops)
  {
  horizontal_tab_stop_list[i+1] = horizontal_tab_stop_list[i] + 8;
  i++;
  }
}

int initialize_vertical_tab_stops()
{
int i;
i = 1;
available_vertical_stops = minimum_vertical_tab_stops;
maximum_vertical_tab_stops = minimum_vertical_tab_stops;

vertical_tab_stop_list[i] = 1;
while(i < maximum_vertical_tab_stops)
  {
  vertical_tab_stop_list[i] = i++;;
  }
}
```

4 GENERAL CONSIDERATIONS

4.1 ALGORITHM FOR HORIZONTAL AND VERTICAL TABS

Setting a stop that is already set or clearing a stop that is already clear has no effect. It is not an error condition to attempt to set more stops than required by the printer architecture.

The upper bound on the number of horizontal and vertical tab stops is implementation-dependent. On devices implementing proportional spacing, tab stops can be set anywhere, not just on character cell boundaries. Conforming devices shall permit, at a minimum:

a. 32 horizontal tab stops

b. 66 vertical tab stops

Conforming software shall not be dependent on a device implementing more than these minimum tab stops to work with all devices conforming to the architecture.

When the number of tab settings is less than the number of possible positions that can be assigned to a tab stop, the algorithm for allocating new tabs shall be as follows.

Starting after the next lower value previous tab stop, each new tab stop value is inserted into the current tab stop list. If more than the allowed number of tab stops have been set, then the highest value previous tab stop is discarded before each new tab stop is entered. Note that if the new tab stop is at the highest position and the allowed number of tab stops have been set, then the new tab stop is ignored.

Algorithm:

```
int horizontal_tabs_6_1(new_horizontal_tab_stop)
int new_horizontal_tab_stop;
{
int i=0, j=0, temp, temp1;

if(pum == SIZE && conformance_level == PSRM_LEVEL_3)
  {
  if(ssu == PIXELS)
    {
    new_horizontal_tab_stop = new_horizontal_tab_stop*120 +
                              line_home_position;
    }
  else if(ssu == DECIPOINTS)
    {
    new_horizontal_tab_stop = new_horizontal_tab_stop*50 +
                              line_home_position;
    }
  else /* Centipoints */
    new_horizontal_tab_stop = new_horizontal_tab_stop*5 +
                              line_home_position;
  }
if(available_horizontal_tab_stops > 0)
  {
  while((new_horizontal_tab_stop < horizontal_tab_stop_list[i]) &&
        (i < maximum_horizontal_tab_stops))
    {
    i++;
    }
if(horizontal_tab_stop_list[i] > 0)
  {
  j = i;
  temp = horizontal_tab_stop_list[j];
  horizontal_tab_stop_list[j] = new_horizontal_tab_stop;
  while((horizontal_tab_stop_list[j] > 0) &&
        (j < maximum_horizontal_tab_stops))
    {
    temp1 = horizontal_tab_stop_list[j+1];
    horizontal_tab_stop_list[j+1] = temp;
    temp = temp1;
    j++;
    }
   }
  }
 }

int vertical_tabs_6_1(new_vertical_tab_stop)
int new_vertical_tab_stop;
{
int i=0, j=0, temp, temp1;
```

```
if(pum == SIZE && conformance_level == PSRM_LEVEL_3)
  {
  if(ssu == PIXELS)
    {
    new_vertical_tab_stop = new_vertical_tab_stop*120 +
                            line_home_position;
    }
  else if(ssu == DECIPOINTS)
    {
    new_vertical_tab_stop = new_vertical_tab_stop*50 +
                            line_home_position;
    }
  else /* Centipoints */
    new_vertical_tab_stop = new_vertical_tab_stop*5 +
                            line_home_position;
  }
if(first_character_flag && pum == CHAR)
   new_vertical_tab_stop = new_vertical_tab_stop + vai;
if(available_vertical_tab_stops > 0)
  {
  while((new_vertical_tab_stop < vertical_tab_stop_list[i]) &&
        (i < maximum_vertical_tab_stops))
    {
    i++;
    }
  if(vertical_tab_stop_list[i] > 0)
    {
    j = i;
    temp = vertical_tab_stop_list[j];
    vertical_tab_stop_list[j] = new_vertical_tab_stop;
    while((vertical_tab_stop_list[j] > 0) &&
          (j < maximum_vertical_tab_stops))
      {
      temp1 = vertical_tab_stop_list[j+1];
      vertical_tab_stop_list[j+1] = temp;
      temp = temp1;
      j++;
      }
    }
  }
}
```

4.2 DEPENDENCIES

4.2.1 Tab Stop Distances

The absolute distance between default tab stops depends on the pitch settings, which in turn can vary due to paper type (8.5 inches × 11 inches or A4 for European paper), orientation (landscape or portrait), and format.

4.2.2 Tab Stop Scaling

DECSHORP and DECVERP are used to scale tab stop positions according to the new pitch settings. No other commands scale the tab stop positions.

4.2.3 HORIZONTAL TAB (HT) Past Last Horizontal Tab Stop

When a HT is received and the active position is past the last horizontal tab stop, the active position moves to the format bound (right margin), setting the right margin flag. Additional HT controls shall not change the active position, even if auto-wrap is enabled. In other words, HT is treated like any other horizontal spacing control function.

4.2.4 VERTICAL TAB (VT) Past Last Vertical Tab Stop

When a VT is received and the active position is past the last vertical tab stop, the active position moves to the Page Home Line (PHL) of the next page, as if a FORM FEED (FF) had occurred. In other words, VT is treated like any other vertical spacing control function.

4.2.5 Position of Tab Stops Relative to the Character Box

The position of a horizontal or vertical tab stop can be specified by the host in the following units.

a. Character cells if POSITIONING UNIT MODE (PUM) is set to CHARACTER. The cell size depends either on the most recent spacing control function (SHS, SVS, SPI, DECSHORP, and DECVERP) or, if the horizontal or vertical parameter of the last occurrence of either SPI, DECSHORP, or DECVERP is 0, the current font. When PUM is set to CHARACTER:

   - The horizontal tab is the left edge of the character box at the baseline.
   - The vertical tab is the top of the character box. Note that this is not the baseline as that depends on the current type size and font and would not be similar to the current character-cell-only devices.
   - Decipoints or pixels, depending on SELECT SIZE UNIT (SSU) and provided PUM is set to SIZE.

   When PUM is set to SIZE:

   - The horizontal tab is the left edge of the character cell at the baseline (the same as when PUM is set to CHARACTER).
   - When PUM is set to SIZE, the vertical tab is the left edge of the character box at the baseline. Note that this is not the top of the character box since the host software positions more accurately when PUM is set to SIZE.

4.2.6 Internal Representation Of Tabs Stops

For character-cell-only printers where PUM is always set to CHARACTER, the tab stops may be stored as bits (one for each possible character cell), or in a tab stop list, depending on implementation.

IMPLEMENTATION GUIDELINE

For printers implementing both settings of PUM, the tab stops should be stored in the smallest unit of measure available to the device. Decipoint, centipoint, and pixel units must be converted to, at least, the smallest common multiple for storage in the tab list. This ensures that scaling of tabs does not introduce errors. See DEC STD 074-6 for more information.

In the case of 300 dots per inch (dpi) printers, this standard strongly recommends the centipoint unit for storage.

5 HORIZONTAL AND VERTICAL TABULATION

5.1 HT - HORIZONTAL TABULATION

```
Levels:  1, 2, 3

Purpose: Move the active position to the next horizontal tab stop.

Format:  HT

7-bit, 8-bit representation:   0/9
```

Description: The HT control character moves the active position to the next horizontal tab stop on the same line. See ANSI X3.4-1977, *American Standard Code for Information Interchange (ASCII)*, and ISO 646-1983, *Information Processing - ISO 7-bit Coded Character Set for Information Interchange*.

NOTES

1. **If the Right Margin (RM) has been reached, additional HT controls shall not cause either motion or wrap, regardless of whether auto-wrap is enabled. If auto-wrap is enabled, wrap occurs when the next graphic character or SPACE is imaged.**
2. **The behavior of HT is not affected by the current setting of PUM.**
3. **If in Justify mode, the text preceding the HT is output unjustified. The text following the HT is justified.**

```
States Affected:

        RIGHT_MARGIN_POSITION: HORIZONTAL_TAB_STOPS:

Algorithm:

int horizontal_tab()
{
while(horizontal_tab_stop_list[active_position.x] <
      right_margin || active_position.x < right_margin)
  {
  active_position.x = horizontal_tab_stop_list[active_position.x + 1];
  if(active_position.x >= right_margin)
     right_margin_flag = TRUE;
  }
}
```

Known Deviations:

- If wrap is enabled, the LA120 wraps tabs immediately
- If wrap is enabled, the LA50 wraps tabs immediately

5.2 VT - VERTICAL TABULATION

```
Levels:  1, 2, 3

Purpose: Move the active position to the next vertical tab stop.

Format:  VT

7-bit, 8-bit:  0/11
```

Description: The VT control character moves the active position to the next line where a vertical tabulation stop is set. See ANSI X3.4-1977 and ISO 646-1983 for more information.

NOTES

1. **VT always sets the First Character Flag under the same conditions that FORM FEED (FF) does. (Refer to DEC STD 074-9 for more information.) Therefore, the baselines of characters will align properly on a vertical tab stop, independent of their actual height.**
2. **If a VT occurs when the active position is below the last set vertical tab, then the active position is advanced to the PHL of the next page and the FCF is set, just as if a FF had occurred. See *DEC STD 074-9 Printer Systems Standard - Positioning Control Functions* for more information.**
3. **VT is not affected by the state of LINE FEED-NEW LINE MODE (LNM) and only moves vertically, as specified in *ANSI X3.64-1979 Additional Control Functions for use with ASCII* (LNM is not in *ISO 6429-1989 Information Processing - Control Functions for 7-bit and 8-bit Coded Character Sets*).**

   Though ISO 646 permits VT to have a CARRIAGE RETURN (CR) movement if prior agreement, ISO 646-1983 deprecates this and prohibits it in *ISO 4873-1986 Information Processing - ISO 8-bit Code for Information Interchange - Structure and Rules for Implementation*.
4. **The behavior of a VT character is not affected by the current setting of PUM or SSU.**
5. **If in Justify mode, the preceding text is output unjustified; the remainder of the line is justified.**
6. **When VT causes an implied FF, the PLU/PLD flag is cleared. The PLU/PLD flag is set whenever a PLU or PLD is executed.**

```
States Affected:

        ACTIVE_POSITION.Y:  POINT_TYPE:

        FIRST_CHARACTER_FLAG:

States/Values affecting (used by) this algorithm:

        BOTTOM_MARGIN_POSITION:  PAGE_HOME LINE:

        VERTICAL_TAB_STOP_LIST:

Algorithm:

int vertical_tab()
{
if(active_position.y > bottom_margin)
  {
  active_position.y = page_home_line;
  first_character_flag = TRUE;
  }
else
  {
  while(vertical_tab_stop_list[active_position.y] <
       bottom_margin ||
       active_position.y < bottom_margin)
    {
    active_position.y = vertical-tab_stop_list[active_position.y+1];
    }
  }
}
```

Known Deviation:

- The LN03 does not advance the active position to the next page if a VT occurs after the last vertical tab stop. However, if the next character is a graphic character or SPACE, the LN03 does advance to the next page because part of the character box would exceed the bottom margin line. Software using upward motion after such a VT would be affected by this LN03 deviation.

6 SETTING HORIZONTAL AND VERTICAL TABULATION STOPS

6.1 HTS - HORIZONTAL TABULATION SET

```
Levels:   1, 2

Purpose:  Set a horizontal tab stop at the active position.

Format:   HTS

8-bit:    8/8
7-bit:    1/11    4/8     (ESC H)
```

Description: The HTS C1 control character sets a horizontal tabulation stop at the active position. See ANSI X3.64-1979 and ISO 6429 for more information.

NOTE

Other horizontal tab stops are not affected by this control function unless all possible settings are already used. Refer to subhead 4.1.

```
States Affected:

       HORIZONTAL_TAB_STOP_LIST: TAB_LIST;

Algorithm:

int hts()
{
horizontal_tabs_6_1(active_position.x);
}
```

Known Deviation: None

6.2 VTS - VERTICAL TABULATION SET

```
Levels:   2

Purpose:  Set a vertical tabulation stop at the active position
          unless the FCF is reset, in which case the tab stop is
          set at the top left corner of the current character box.

Format:   VTS

8-bit:    8/10
7-bit:    1/11    4/10    (ESC J)
```

Description: The VTS C1 control character sets a vertical tabulation stop at the active line. See ANSI X3.64-1979 and ISO 6429.

```
States Affected:

        VERTICAL_TAB_STOP_LIST: TAB_LIST;

Algorithm:

int vts()
{
vertical_tabs_6_1(active_position.y);
}
```

Known Deviation: None

6.3 DECHTS - HORIZONTAL TABULATION SET

```
Levels:   2
Purpose:  Set a horizontal tab stop at the active position.
Format:         ESC     1
7-bit, 8-bit:  1/11    3/1
```

Description: The DECHTS escape sequence sets a horizontal tabulation stop at the active position. See *DEC STD 138-0 Registry of Control Functions for Character Imaging Devices* for more information.

NOTES

1. **None of the other horizontal tab stops are affected by this control function, unless all possible settings are already used. Refer to subhead 4.1.**
2. **DECVTS is identical to the ANSI/ISO VTS C1 control character. Therefore, conforming software shall use the ANSI/ISO VTS, rather than DECVTS.**

```
States Affected:
        HORIZONTAL_TAB_STOP_LIST: TAB_LIST;
Algorithm:
int dechts()
{
horizontal_tabs_6_1(active_position.x);
}
```

Known Deviation: None

6.4 DECVTS - VERTICAL TABULATION SET

```
Levels:     2

Purpose:    Set a vertical tabulation stop at the active position
            unless the FCF is reset, in which case the tab stop is
            set at the top left corner of the current character box.

Format:         ESC     3

7-bit, 8-bit:  1/11     3/3
```

Description: The DECVTS escape sequence sets a vertical tabulation stop at the active line. See DEC STD 138-0 for more information.

NOTE

DECVTS is identical to the ANSI/ISO VTS C1 control character. Therefore, conforming software shall use the ANSI/ISO VTS, rather than DECVTS.

```
States Affected:

        VERTICAL_TAB_STOP_LIST: TAB_LIST;

Algorithm:

int decvts()
{
vertical_tabs_6_1(active_position.y);
}
```

Known Deviation: None

6.5 DECSHTS - SET HORIZONTAL TABULATION STOPS

```
Levels:  2

Purpose: Set several horizontal tabs at the positions specified
         by a single command.

Format:        CSI   Pn...   ;    Pn...  u     Default Pn: none

8-bit:         9/11  Pn    3/11   Pn... 7/5
7-bit:    ESC  5/11  Pn    3/11   Pn... 7/5
```

Description: The DECSHTS control sequence sets a horizontal tabulation stop at each position corresponding to each parameter value.

The units in which each Pn is expressed depend on the settings of the PUM and SSU.

There are a variable number of numeric parameters, but no default value. See DEC STD 138-0 for more information.

NOTES

1. **DECSHTS does NOT clear other horizontal tabulation stops as long as there is sufficient space available for these stops. Refer to subhead 5.1.**
2. **DECSHTS is identical to the ISO HORIZONTAL TABULATION SET ABSOLUTE (HTSA) (ISO 6429-1989), except that HTSA clears all existing horizontal tabulation stops first, while DECSHTS does not.**
3. **The architecture requires conforming devices to implement 16 parameter values for a control sequence. Therefore, conforming software shall not send more than 16 parameters. If more than 16 tab stops are needed, either send another DECSHTS or use HTS.**

```
States Affected:

        HORIZONTAL_TAB_STOP_LIST: TAB_LIST;

Algorithm:

int decshts()
{
int i = 0;

/* get parameter list from parser */
while(new_horizontal_tab_stop_list[i] > 0)
  {
  horizontal_tabs_6_1(new_horizontal_tab_stop_list[i]);
  i++;
  }
}
```

Known Deviation: None

6.6 DECSVTS - SET VERTICAL TABULATION STOPS

```
Levels:   2

Purpose:  Set several vertical tabs at the positions specified
          by a single command.

Format:          CSI     Pn...   ;       Pn...   v    Default Pn: none

8-bit:           9/11    Pn...   3/11    Pn...   7/6
7-bit:       ESC 5/11    Pn...   3/11    Pn...   7/6
```

Description: The DECSVTS control sequence sets a vertical tabulation stop at a position corresponding to each parameter value. See DEC STD 138-0 for more information.

NOTES

1. **DECSVTS does NOT clear other vertical tabulation stops as long as sufficient space is available for these steps. Refer to subhead 5.1.**
2. **There is no similar ANSI/ISO control sequence to DECSVTS (unlike DECSHTS, for which ANSI/ISO HTSA is similar).**
3. **The architecture requires conforming devices to implement 16 parameter values for a control sequence. Therefore, conforming software shall not send more than 16 parameters. If more than 16 tab stops are needed, send another DECSVTS.**

States Affected:

```
        VERTICAL_TAB_STOP_LIST: TAB_LIST;
```

Algorithm:

```
int decsvts()
{
int i = 0;

/* get parameter list from parser */
while(new_vertical_tab_stop_list[i] > 0)
  {
  vertical_tabs_6_1(new_vertical_tab_stop_list[i]);
  i++;
  }
}
```

Known Deviation: None

7 CLEARING HORIZONTAL AND VERTICAL TABULATION STOPS

7.1 TBC - TABULATION CLEAR

```
Levels: 1, 2

Purpose:  Clear horizontal and/or vertical tab stops.

Format:          CSI     Ps      g          Default Ps:0

8-bit:           9/11    Ps      6/7
7-bit:      1/11 5/11    Ps      6/7
```

Description: The TBC control sequence clears horizontal and/or vertical tabulation stops as indicated by the selective parameter. See ANSI 3.64-1979 and ISO 6429-1989.

Ps	Effect
[omitted]	Clear horizontal tab stop at the active position
0	Clear horizontal tab stop at the active position
1	Clear vertical tab stop at the active line
2	Clear all horizontal tab stops in the active line
3	Clear all horizontal tab stops in the page
4	Clear all vertical tab stops
[> 4]	Reserved for future standardization

NOTES

1. **There is no difference between selections 2 and 3, as the ISO 6429 TABULATION STOP MODE (TSM) is permanently set to MULTIPLE.**
2. **Like all other ANSI selective parameter sequences, (Ps) parameters can be grouped into one sequence by separating the parameters with a semicolon (3/11).**

```
States Affected:

        HORIZONTAL_TAB_STOP_LIST: TAB_LIST;
        VERTICAL_TAB_STOP_LIST: TAB_LIST;

Algorithm:

int tabulation_clear()
{
int i, ps;

switch(ps)
  {
  case 0: horizontal_tab_stop_list[active_position.x] = 0;
          break;
  case 1: vertical_tab_stop_list[active_position.y] = 0;
          break;
  case 2:
  case 3:
          for(i=1; i < maximum_horizontal_tab_stops; i++)
              horizontal_tab_stop_list[i] = 0;
          break;
   case 4:
          for(i=1; i < maximum_vertical_tab_stops; i++)
              vertical_tab_stop_list[i] = 0;
          break;
   }
 }
```

Known Deviation: None

7.2 DECCAHT - CLEAR ALL HORIZONTAL TABULATION STOPS

```
Levels:   2

Purpose:  Remove all horizontal tab stops from the horizontal tab
          stop list.

Format:                   ESC     0

7-bit, 8-bit:             1/11    3/0
```

Description: The DECCAHT escape sequence clears all horizontal tab stops.

```
States Affected:

        HORIZONTAL_TAB_STOP_LIST: TAB_LIST;

Algorithm:

int deccaht()
{
int i = 0;

for(i=1;maximum_horizontal_tab_stops;i++)
    horizontal_tab_stop_list[i] = 0;
}
```

Known Deviation: None

7.3 DECCAVT - CLEAR ALL VERTICAL TABULATION STOPS

```
Levels:   1, 2

Purpose:  Remove all vertical tab settings from the vertical tab
list.

Format:           ESC     4

7-bit, 8-bit:     1/11    3/4
```

Description: The DECCAVT escape sequence clears all vertical tabulation stops.

```
States Affected:

        VERTICAL_TAB_STOP_LIST: TAB_LIST;

Algorithm:

int deccavt()
{
int i = 0;

for(i = 1;i<maximum_vertical_tab_stops;i++)
    vertical_tab_stops_list[i] = 0;
}
```

Known Deviation: The LN01 does not implement DECCAVT.

APPENDIX A REFERENCED DOCUMENTS

A.1 EL-Class Digital Documents

EL-Class Number	Document Title
EL-00074-06	*DEC STD 074-6 Printer Systems Standard - Page Coordinate System*
EL-00074-09	*DEC STD 074-9 Printer Systems Standard - Positioning Control Functions*
EL-00138-00	*DEC STD 138-0 Registry of Control Functions for Character Imaging Devices*

Use VTX SMC to order copies of EL-class documents from Standards and Methods Control. Send distribution questions to JOKUR::SMC or call DTN: 234-4423.

A.2 Other Documents

Document Number	Document Title
ANSI X3.4-1977	*American Standard Code for Information Interchange (ASCII)*
ANSI X3.64-1979	*Additional Control Functions for use with ASCII*
ISO 4873-1986	*Information Processing - ISO 8-bit Code for Information Interchange - Structure and Rules for Implementation*
ISO 6429-1989	*Information Processing - Control Functions for 7-bit and 8-bit Coded Character Sets*
ISO 646-1983	*Information Processing - ISO 7-bit Coded Character Set for Information Interchange*

Note: All control function acronyms that do not contain a "DEC" prefix are drawn from these standards.

A.3 Ordering Information

Following are the sources for documents not available from Standards and Methods Control.

The following ANSI and ISO standards are adhered to in this specification. The descriptions of control functions in this standard use the exact language from the applicable standard. Additional Digital-specific information is included in notes following the description.

- ANSI, IEC, and ISO Documents
  American National Standards Institute (ANSI),
  1430 Broadway, New York, NY 10018,
  Telephone (212) 354-3300 x479

```
+-------------------------------------------------------------------+
|                        READER COMMENTS                            |
|                                                                   |
| Your  comments  and suggestions  will help  Standards and Methods |
| Control improve their services and documents.                     |
+-------------------------------------------------------------------+
```

Did you request this document? _____ If so, did it arrive within a satisfactory period of time? _____ Please comment.

What are your impressions of this document? Consider format, organization, completeness, readability, and illustrations.

————————FOLD ON THIS LINE———————————————————— -

Did you find technical or clerical errors in this document? If so, please specify the page number(s) and the error(s).

Are the instructions for the update package clear? _____
Was an index available? _____ If not, is one needed? _____
Do you have other suggestions for improving this document?

The following information is optional:

Name _____ Mailstop _____

Department _____ Node _____

Send your comments to JOKUR::PROJECTS, or fold, staple, and send this page through interoffice mail to:

```
+------------------------------+
|       READERS' COMMENTS      |
| STANDARDS AND METHODS CONTROL |
|           NRO4/D4            |
+------------------------------+
```